**A PROJECT REPORT (IBM 19 G-17)**

**ON**

**PLANT DISEASE PREDICTOR**

*A report submitted in partial fulfilment of the requirement for the award of*

*The degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

A picture containing drawing

Description automatically generated

**Submitted to:**

Ms. Meghavi Rana

Assistant Professor

**Submitted by:**

Anisha Srivastava-190110021

Priyanshu Chauhan-190110022

Surbhi Singh-190110023

Ujjwal Tayal-190110024

**SCHOOL OF COMPUTING**

**DIT UNIVERSITY, DEHRADUN**

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

**Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.**

**CANDIDATES DECLARATION**

I hereby certify that the work, which is being presented in the Report, entitled **Plant Disease** **Predictor,** in partial fulfilment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of my work carried out during the period from ***26 August 2021*** to ***1 November 2021*** under the guidance of Ms. Meghavi Rana.

**DATE:**  **16/10/2021**

|  |  |
| --- | --- |
| **Anisha Srivastava** |  |
| **Priyanshu Chauhan** |  |
| **Surbhi Singh** |  |
| **Ujjwal Tayal** |  |

**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible to complete this without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

First, we are thankful to our project guide Ms. Meghavi Rana under whose guideline we were able to make progress in our project. We are whole heartedly thankful to her for giving us her valuable time & attention & for providing us a systematic way for progressing our project in time.

Then we are thankful to all members of this project, without their support and hardworking nature, it would have been impossible to make any progress in this project.

Anisha Srivastava

Priyanshu Chauhan

Surbhi Singh

Ujjwal Tayal

**ABSTRACT**

Machine Learning Machine is one of the branches of Artificial Intelligence to work automatically or give instructions to a specific action plan. The objective of ML is to understand the structure of data and to integrate that data into models that can be understood and employed or used by humans to get some result. The proposed experimental task is to analyze the algorithms of the complex machines that applied to predict plant disease. If a sample displays the material assets of disease, as a response to the pathogen, it can be cured and hence crop loss can be minimized. Visual features such as shape, size, spots, suspension, certainly help to combat the condition. This design works with all the same features and uses technology to find an issue that will be seen in the future.**TABLE OF CONTENT**

CHAPTER PAGE No.

Candidate’s Declaration 2

Acknowledgement 3

Abstract 4

Chapter 1 -- Introduction 8-18

* 1. Machine Learning 10-12
  2. Deep Learning 12-18

Chapter 2 –Project Description 19-26

2.1. Purpose 19-20

2.2 Overall system block diagram 21

Chapter 3 –Tools and Technologies

**Example – Hardware and Software**

Chapter 4- Implementation Modules and Screen Shots

Chapter 5 – Conclusion & Road map of Phase-2

Bibliography

**Plagiarism report 20% only**

**LIST OF TABLES**

TABLE NAME PAGENO.

1 – Machine learning vs deep learning 18

2 – CNN vs ANN 28

**CHAPTER 1 – INTRODUCTION**

Plant diseases are a major threat to food security, and rapid detection remains difficult in many parts of the world due to a lack of proper system and information. The combination of the global expansion of smartphones and devices as well as advances in computer visualization made possible through in-depth study has opened the way for diagnostic-assisted diagnostic diseases. Using the public database of diseased and healthy plant leaves, we train a deep convolutional neural network to identify plant species and diseases. A set of trained models provides accurate results, indicating the feasibility of this approach. The method of training for in-depth learning models is in high demand and the publicly available image data sets provide a clear diagnosis of plant diseases worldwide.

Machine learning behaves like a self-study concept that will work without human interference. Now self-driving cars of the day, handwriting awareness, stock market are some examples of machine learning concepts. Machine learning will be able to predict the future based on past or historical details.

This project will not only predict plant disease but will also act as an interactive web application that will help users to seek information about various plants, their diseases, and their cure. An example of a plant disease dataset is shown below:



Fig 1.1 Sample of dataset

* 1. **Machine Learning**

Machine learning is an Artificial Intelligence application advocated lately. It is the innovation of showing a machine, here, PC, to learn explicit things with no point by point or explicit programming. By and large, it centres around recovering information and gaining from it. The information can be breaking down utilizing different calculations and afterward the yield is anticipated and produced. Machine learning is exceptionally helpful when-

* Analysing enormous informational indexes and producing yield
* We need an ideal yield informational collection.

We feed a preparation informational collection and get a yield as per similitudes, designs, and so forth.

* We need human-like knowledge from our machine

There are two fundamental sorts of Machine Learning-

**1.1.1 Supervised Learning** – As the name proposes, in this sort of learning we regulate a machine by giving article (info) and names (yield) to a machine. The information is known as Training Data Set. The calculations can check their answers in this kind of figuring out how to acquire an advanced yield.

Where you have input factors (X) and a yield variable (Y), and you utilize a calculation to take in the planning capacity from the contribution to the yield as appeared in condition 1 is Supervised Learning.

(1) The objective is to surmise the planning capacity so well that when you have new info information (x) that you can foresee the yield factors (Y) for that information.

Managed learning discovers its applications in money-related fields for scoring credit, algorithmic exchanging, and security arrangement; in the organic field for the location of tumours and medication revelation; in the energy field for cost and burden determining, and in design acknowledgment applications for discourse and pictures acknowledgment.

Administered learning issues can be additionally gathered into relapse and arrangement issues.

1. **Classification**: A grouping issue is the point at which the yield variable is a classification, for example, "yes" or "no", "green" or "blue" or "malady" and "no illness".

Characterization prescient demonstrating is the undertaking of approximating a planning capacity (f) from input factors (X) to discrete yield factors (y).

The yield factors are regularly called names or classifications. The planning capacity predicts the class or classification for a given perception. For instance, an email or text can be named having a place with one of two classes: "spam" and "not spam".

* A grouping issue necessitates that the order of models is done into one of at least two classes.
* A grouping can have genuine esteemed or discrete information factors.
* An issue with two classes is regularly called a two-class or parallel order issue.
* An issue with multiple classes is regularly called a multi-class arrangement

Issue.

* An issue where a model is relegated to different classes is known as a multi-mark characterization issue.

Probably the most utilized characterization calculations: Random Forest, Decision Trees Logistic, Regression, Support Vector Machines

1. **Regression**: A relapse issue is a point at which the yield variable is a genuine worth, for example, "dollars" or "weight".

Relapse prescient displaying is the errand of approximating a planning capacity (f) from input factors (X) to a constant yield variable (y).

A consistent yield variable is a genuine worth, for example, a whole number or gliding point esteem. These are regular amounts, for example, sums and sizes.

For instance, a house might be anticipated to sell for rupee esteem, maybe in the scope of Rs.10, 00,000 to Rs.20, 00,000.

* A relapse issue requires an amount of expectation.
* A relapse can have either genuine esteemed or discrete information factors.
* An issue with various information factors is frequently called a multivariate relapse issue.
* A relapse issue where input factors are requested by time is known as a period arrangement gauging issue.

Since a relapse prescient model predicts an amount, the model ability must be accounted for as a mistake in those expectations. Probably the most utilized Regression models:

1. Linear Regression
2. Decision Tree Regression
3. Polynomial Regression

* 1. **Deep Learning**

In-depth learning is a phase of AI strategies that use multiple layers of indirect data management for controlled or solo element extraction and transformation or systematic and systematic research. It contains many different layers of indirect data capture, in which a low-level view helps to express very high-level ideas.

The in-depth artificial neural network is not suitable for the care of many complex information, which is reflected in many common applications, for example, normal speech, images, data retrieval, and other human-like data management applications. In-depth study is appropriate for such applications. With in-depth learning, it is possible to visualize, mimic, and edit the structure of machine information with similar attempts. Google is a pioneer in deep understanding analysis, initiated by Andrew Ng.

In-depth learning provides a multi-line management of human testing in shallow engineering. The most important idea for in-depth learning is to use a variety of balanced management and apply multiple layers of engineering. The engineering layers are technically advanced. After previous training is completed, the input of each layer goes to its nearest layer and continues to provide output. Mostly, previous training for the selected layer was done in an unsupervised manner and later this previous training for the required data details.

In-depth learning follows a broad approach to dealing with big data management. This approach allows for the production of information to consider different factors, different time, and different levels. In-depth learning encourages the process and processing of information into different layers as it is now the right time (event), its scale, or the environment.

Deep learning is regularly connected with counterfeit neural organizations. There are three classifications of profound learning designs:

* Generative
* Discriminative
* Hybrid Deep learning architecture

Architectures from general classifications centre on the pre-preparing of a layer in a solo manner. This methodology kills the trouble of preparing lower-level models, which depend on the past layers. Each layer can be pre-prepared and later remembered for the model for additional overall tuning and learning. Doing these purposes, the issue of preparing neural organization design with numerous layers and empowers deep learning.

The construction of a Neural network may have the power to discriminate by placing a stack on each layer with initial data or with a different data combination and then building a deeper learning structure. The descriptive model always considers the yield of the neural network as a restrictive allocation to all thought scores for the thought-provoking data collection, which will be further enhanced by objective work. Crossover engineering incorporates structural design and discriminatory design properties. Generally, one can do in-depth study in the following way-

• Create an organization that contains a knowledge base and a covered background with key harps

• Train the organization

• Add another hidden layer over the previously read network to create a new network

• Re-train the network

• Re-add more layers and after every installation, re-train the network

**1.2.1 Different types of deep learning models**

**a) Auto-encoders**

Auto-encoder is a fake neural organization equipped to read various coding designs. The exact type of auto-encoder is very similar to a multilayer perceptron, which contains a layer of information or at least one hidden layer, or a production layer. The critical difference between the run-of-the-mill multilayer perceptron and the feedforward neural organization and auto-encoder in the number of harps in the yield layer. Thanks to the auto-encoder, the yield layer contains the same amount of hubs as in the information path. Instead of expecting objective attributes depending on the vector yield, the auto-encoder needs to preview its sources. A comprehensive study tool plan is as follows.

For each detail x,

• Perform a feed transfer to obtain the operating capacity given to all hidden layers and yield layers.

• Identify deviations between determined attributes and sources using appropriate error function

• Make a mistake back to reload loads

• Repeat the delivery until the harvest is acceptable.

If the number of hubs in the folded layers is below the knowledge / production areas, then the creation of the last hidden layer is considered a complete representation of the data sources. At a time when hidden layer domains are beyond the knowledge base, the auto-encoder may become accustomed to human activity and become useless in most cases.

b) Deep Belief

The deep-seated belief network is the answer to the problem of dealing with the limited intensity of the target and the surrounding fields while using a standard multilayer perceptron. It is a type of deep depth selection that involves multiple layers of non-functional objects and relationships between layers. The deep condemnation organization can be seen as a machine bound in Boltzmann (RBM), where the hidden layer of each subnetwork moves as the open information of the organization layer. It makes the visual layer significantly reduced to a set of preparation for the boring organizational layer. In line with these lines, each layer of the organization is prepared independently and greedily. Slab contraction joints should intersect at the openings for columns. An account of the preparation of the deep-seated network is provided by the following:

• Consider the vector of data sources

• Train Boltzmann's locked machine using an information vector and get a weight lattice

• Train the two lower layers of the organization using this weightlifting framework

• Generate a vector of new information using an organization (RBM) by testing or referring to the creation of hidden units

• Repeat the process until two layers of organization are reached

Deep network measurement is basically the same as a multilayer perceptron. Such deep-seated belief networks help to show acoustic

**c) Convolutional Neural Networks**

Another variation of the feed-forward multilayer perceptron network is the convolutional neural network (CNN). It is a type of neural organization, in which individual neurons are asked to respond to all areas in the visual environment.

Deep CNN works by continuously displaying captions for information and integrating them into the deepest part of the organization. One way to understand yourself is that the larger layer will try to see the edges and structures of the acquisition structure. After that, the layers that will come out will try to join them into simple shapes and over time become the structures of various objects, lighting, scales, and so on. The final layers will include a detailed image with all the structures, and the final forecast is the same as the weight of them all. Therefore, deep CNNs can show complex forms and behaviors, providing accurate predictions.

Such an organization follows a system of biological images. Phones in this graphic cortex touch small areas under the image field, called the answering field. The sub-regions are designed to understand all the visible space, and the cells move as close channels to the information space. Background calculation is used to adjust the parameters of all parts of the solution. Moreover, all the pieces are repeated over the whole picture with the same parameters. There are convolutional managers who remove the remarkable prominence of information. In addition to the convolutional layer, the organization consists of a layer of a straightforward unit, combining layers to obtain the maximum or standard of the object above the image area, and a frame of misfortune that contains the use of explicit misfortune. Image approval and video surveillance and common language handling are important uses for such a neural organization.

The PC vision region has seen continuous improvement in the past in almost any age. One of the most popular shows is CNN. Currently, CNN’s in-depth design is a center for seeing more complex PC objects and pattern applications, for example, self-driving cars, motion recognition, automatic tagging of collections in our Facebook photos, facial recognition, and custom number recognition.

**d) Recurrent Neural Networks**

The convolutional model takes a number from the established sources, creating a modified vector as a result of the number of steps described earlier. Part-time organizations allow us to work beyond the collection of veggies with information and products. Due to the repetitive neural organization, the interaction between units forms an integrated cycle. In contrast to a typical neural organization, intermediate and product organization details are not free but highly related. In addition, the repetitive neural organization shares common parameters in each layer. One can set up a repetitive organization in the same way as a neural culture organization using a regression strategy.

Here, the slope limitations do not depend on the current prediction but on past development as well. Variations called bidirectional intermittent neural organization are used for other systems. The bidirectional neural organization looks at the past and the general yield of the future. In organizations with two occasional straight lines, in-depth learning can be achieved by introducing more subtle layers. Such deep organizations offer higher learning limits with a wealth of reading material. Conversation, image editing, and general language management are part of the application areas where this model can be beneficial.

**1.2.2 Reinforcement Learning to Neural Networks**

This learning is a sort of dynamic programming and directed learning. Average segments of the methodology are climate, operator, activities, strategy, and cost capacities. The specialist goes about as a regulator of the framework; strategy decides the moves to be made, and the prize capacity determines the general target of the fortification learning issue. A specialist, accepting the most extreme conceivable prize, can be viewed as playing out the best activity for a given state.

Here, an operator alludes to a theoretical substance, either an item or a subject (independent vehicles, robots, people, client service talk bots, and so on.), which takes part in some activity. The condition of an operator alludes to its position and condition of being in its theoretical climate, for instance, a particular situation in a computer-generated experience world, a structure, or the position and speed on a course. Profound support learning holds the assurance of a summed-up learning method that can easily learn valuable conduct with almost no criticism. It is an energizing and testing region, which will without a doubt be a basic aspect of things to come AI scene.

**Table 1**- Machine Learning vs. Deep Learning

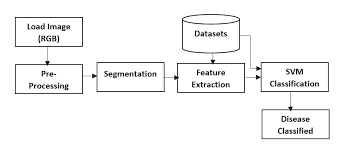
|  |  |  |
| --- | --- | --- |
| **S.NO** | **Machine learning** | **Deep learning** |
| 1. | Can be defined as a shallow neural network which consists of one input and one output, with barely one hidden layer. | To be qualified for deep learning, there must be at least three layers. |
| 2. | Requires small amount of data. | Requires large amount of unlabelled training data. |
| 3. | This technique requires labelled parameters to perform feature extraction. | This technique performs automatic feature extraction without any human intervention requirement. |
| 4. | High-performance hardware is not required. | High-performance hardware is required. |
| 5. | Needs accurately identified features by human intervention. | Can create new features. |
| 6. | Tasks are divided into small portions and then forms a combined effect. | Offers end-to-end problem solution. |
| 7. | Takes less time to train. | Takes a lot of time to train. |

**CHAPTER-2 PROJECT DESCRIPTION**

**2.1 Purpose**

The main purpose of the project is to predict plant disease. For this, we will be making a machine learning model using Convolution Neural Network (CNN). The model will read one image at a time and recognize various patterns or spots present on the leaf which may be because of some disease. The data set of healthy as well as unhealthy plants will be used for training and testing purposes. At last, we will prepare a GUI (graphical user interface).

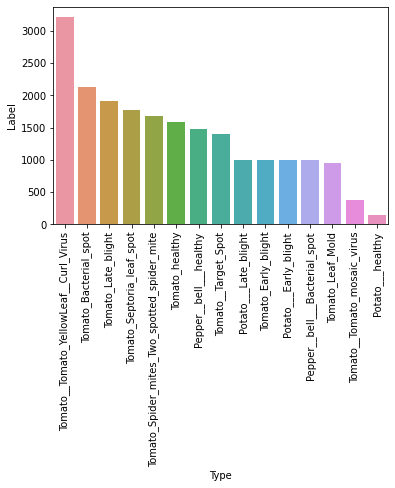
Every year, plant disease causes a significant loss of valuable food crops around the world. The plant and crop disease management practices implemented to reduce damages have changed considerably with time. Today, through the application of new information and technologies, it is possible to predict or change the severity of disease using modern data analysis techniques.



**Fig 2.1 –** Steps to predict disease

1. **Dataset**

We used the Village Dataset. The Plant Village database contains 42,000 healthy and unhealthy foliage classified into various categories by disease. We analyzed more than 40,000 images of plant leaves with distributed labels from various categories and tried to predict the disease class. We will make the new size of each image 256 × 256 pixels and use it to optimize and predict the model in this compressed image.



**Fig 2.2** Number of images present in dataset

1. **Data Processing and Augmentation**

Picture augmentation performs a key function in constructing an effective picture classifier. although datasets may additionally contain masses to a couple of thousand training examples, the variety or quantity may nevertheless now not be sufficient to construct an correct model. some of the various image augmentation options are flipping the picture vertically or horizontally, rotating via various angles, and scaling the photo. these augmentations assist boom the applicable information within the dataset. the scale of every photo in the Plant Village dataset is observed to be 256 x 256 px. The records pre-processing and photo augmentation are performed the usage of the Kera’s, a deep-getting to know framework. The augmentation alternatives used for education are as followed:

• Rotation – Will rotate a schooling photo randomly over numerous angle.

• Brightness - helps the version to conform to a version in lighting fixtures even as feeding photographs of various brightness at some point of training

• Shear - adjust the shearing angle of the picture

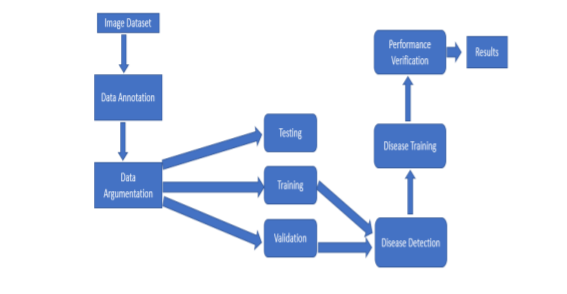
1. **Basic system block diagram**

The general procedure is split into 3 tiers:

1. Enter photos are first created with the aid of Android gadgets or uploaded to our web utility with the aid of users.

2.Segmentation pre-processing includes the system of image segmentation, picture enhancement, and color area conversion. First, the photograph is more advantageous with a clear out. Then every photo is transformed into an array. every photograph call is converted to a binary subject.

3. CNN version is skilled to perceive illnesses in every plant elegance. outcomes located in stage 2 are used to call up a classifier, which is educated to categorise various sicknesses in that plant image. If not present, the leaves are classified as "healthy".

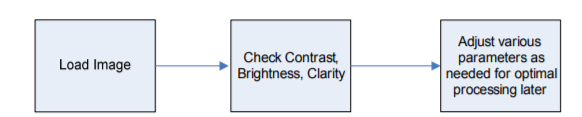


**Fig.2.3 Flowchart of disease detection**

**2.3 Functional description**

In functional description we will talk about three main topics regarding this project which are listed below:

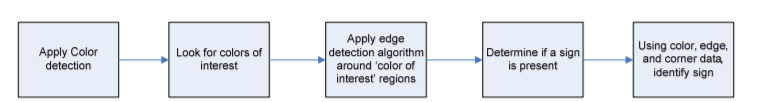
1. Pre-processing block
2. Image processing and recognition
3. Highlights/output

**2.3.1 – Pre-processing block**

**Figure 2.4** – Pre-processing block

Pre-processing will stack the picture just as check differentiation & brilliance. Figure 2.4 shows the progression of the pre-processing block. If these boundaries are off from our ideal qualities for these, modifications will be performed. This will permit the planning group to have the option to guarantee that the picture is reasonable for preparing. On the off chance that the product can't acquire the difference required, it will most likely be unable to distinguish if there is a disease in the picture.

**2.3.2 – Image processing and recognition**

****

**Fig.2.5: Image processing**

Fig. 1. Sample images in the dataset [6].

3.2 Data preprocessing and feature extraction

Data preprocessing is important task in any computer vision-based system. Fig. 2

illustrates the preprocessing steps for each image. To get precise results, some

background noise should be removed before extraction of features. So first the RGB

image is converted to greyscale and then Gaussian filter is used for smoothening of the

image. Then to binaries the image, Otsu’s thresholding algorithm is implemented. Then

morphological transform is applied on binarized image to close the small holes in the

foreground part. Now after foreground detection, the bitwise AND operation on

binarized image and original color image is performed to get RGB image of segmented

leaf. Now after image segmentation shape, texture and color features are extracted from

the image. By using contours, area of the leaf and perimeter of the leaf is calculated.

Contours are the line that joins all the points along the edges of objects having same

color or intensity. Mean and standard deviation of each channel in RGB image is also

estimated. To obtain amount of green color in the image, image is first converted to

HSV color space and we have calculated the ratio of number of pixels having pixel

intensity of hue (H) channel in between 30 and 70 and total number of pixels in one

channel. Non-green part of image is calculated by subtracting green color part from 1.

Fig. 1. Sample images in the dataset [6].

3.2 Data preprocessing and feature extraction

Data preprocessing is important task in any computer vision-based system. Fig. 2

illustrates the preprocessing steps for each image. To get precise results, some

background noise should be removed before extraction of features. So first the RGB

image is converted to greyscale and then Gaussian filter is used for smoothening of the

image. Then to binaries the image, Otsu’s thresholding algorithm is implemented. Then

morphological transform is applied on binarised image to close the small holes in the

foreground part. Now after foreground detection, the bitwise AND operation on

binarised image and original color image is performed to get RGB image of segmented

leaf. Now after image segmentation shape, texture and color features are extracted from

the image. By using contours, area of the leaf and perimeter of the leaf is calculated.

Contours are the line that joins all the points along the edges of objects having same

color or intensity. Mean and standard deviation of each channel in RGB image is also

estimated. To obtain amount of green color in the image, image is first converted to

HSV color space and we have calculated the ratio of number of pixels having pixel

intensity of hue (H) channel in between 30 and 70 and total number of pixels in one

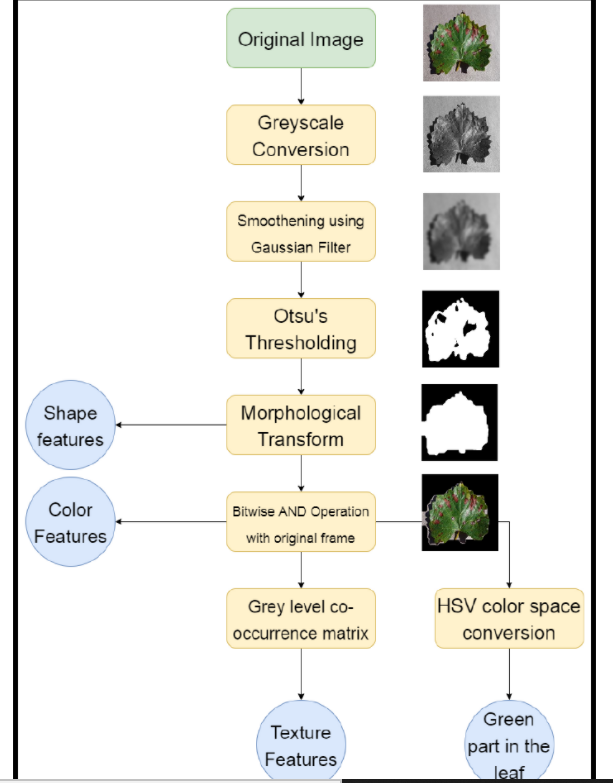
channel. Non-green part of image is calculated by subtracting green color part from 1.

Data processing is an essential function in any computer-based system. To get direct results, some background noise should be removed before removing features and guessing results. So, initially, the RGB image is converted to greyscale, and then the filter is used to zoom the image.

Now after the status separation of the image separation, texture, and color symbols are removed from the image. Using boundaries, leaf area and leaf rotation are calculated.

Lines are lines that connect all the points at the edges of objects of the same color or intensity. The mean and standard deviation (SD) of each channel in the image is also calculated. To get the green color value in the image, the image is first converted to HSV color

The non-green part of the image is calculated by subtracting the green part from 1.



After extracting color features from the image, we have extracted texture features from

grey level co-occurrence matrix (GLCM) of the image [7].

Fig. 2. Steps for data preprocessing and feature extraction.

**Fig.2.6** Steps for data preprocessing and feature extraction.

Fig. 1. Sample images in the dataset [6].

3.2 Data preprocessing and feature extraction

Data preprocessing is important task in any computer vision-based system. Fig. 2

illustrates the preprocessing steps for each image. To get precise results, some

background noise should be removed before extraction of features. So first the RGB

image is converted to greyscale and then Gaussian filter is used for smoothening of the

image. Then to binaries the image, Otsu’s thresholding algorithm is implemented. Then

morphological transform is applied on binarised image to close the small holes in the

foreground part. Now after foreground detection, the bitwise AND operation on

binarised image and original color image is performed to get RGB image of segmented

leaf. Now after image segmentation shape, texture and color features are extracted from

the image. By using contours, area of the leaf and perimeter of the leaf is calculated.

Contours are the line that joins all the points along the edges of objects having same

color or intensity. Mean and standard deviation of each channel in RGB image is also

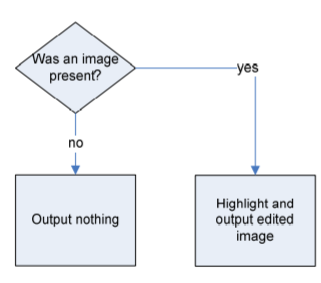
estimated. To obtain amount of green color in the image, image is first converted to

HSV color space and we have calculated the ratio of number of pixels having pixel

intensity of hue (H) channel in between 30 and 70 and total number of pixels in one

channel. Non-green part of image is calculated by subtracting green color part from 1.

**2.3.3 – Highlights/output**



**Figure 2.7** – Highlights/output

This subsystem will take the information and make the final output. Output sub system is appeared in Figure 2.7 which shows the progression of the square. Without a pattern recognized, nothing will be yield.

**2.4 – Functional requirements**

Pre-processing will check difference, splendour and lucidity. This square will make sure that the picture is prepared to have picture handling done to it. In the wake of going through this pre-handling block, the image will be prepared to have preparing calculations applied to it. The utilization of handling calculations will take the pre-prepared picture and discover shades and search for shapes identifying with the spots or patterns we are looking for.

**CHAPTER-3 TOOLS AND TECHNOLOGIES**

**3.1 Hardware requirements**

1- Intel core i5

2- Graphics card 2 GB

3- RAM 8 GB

**3.2 Software Requirements**

1 - Windows operating system

2 – VS Code

3 – NumPy

4 – Flask

5 – Keres

**a) VS Code**

Visual Studio Code is an Integrated Development Environment (IDE) made by Microsoft for Windows, Linux and macOS. Features can include debugging support, syntax highlighting, intelligent coding, reuse code, and embedded Git. Administrator can change IDE themes as per his mood, shortcuts (keyboard), preferences, and install extensions that add additional functionality as per usage. Various extensions can be installed in VS Code.

**b) Flask**

Flask is a microframework written in Python language. It is classified as a microframework because it doesn’t require any library. It has no layer to abstract data from database, or any other components where pre-existing libraries which provide common functionality. However, Flask supports extensions that can add application features, as if they are being implemented in Flask itself.

**c) Keras**

Keras contains a wide variety of neural-network building squares, for example, layers, objectives, capabilities, directional agents, and a large group of tools to make working with image and text details easier to disrupt the writing base by designing a deep neural organizational code. The code has been simplified on GitHub, and network support conversations include the GitHub problem page, and the Black channel.

In addition to normal neural associations, Keres has supported dynamic and intermediate neural associations. Maintains other common usage layers such as stop, group rate, and integration.

Keres allows clients to produce in-depth models on mobile, web, or Java Virtual Machine. Similarly it allows for the use of distributed modifications of deep learning models in GPU.

**CHAPTER – 4 IMPLEMENTATION MODULES AND SCREENSHOTS**

**4.1 Proposed CNN Model**

The steps of development of CNN model are described below.

**4.1.1 Convolution Neural Network (CNN)**

Convolutional neural networks (CNN) are one of the most well-known models used in todays world. This model uses multiple layers for perceptions and consist of at least one convolutional layer that can be either altogether be associated or pooled. These convolutional layers may include maps that record an area of picture which is eventually broken into square shapes and conveyed for nonlinear preparing.

Advantages:

* Great exactness in picture acknowledgment issues.
* Automatically identifies the significant highlights with no human efforts.
* Weight sharing.

Disadvantages:

* CNN don't encode the position and direction of item.
* Lack of capacity to be spatially invariant to the information.
* Lots of preparing information is required

**Table -2** Convolutional neural network vs. Artificial neural network

|  |  |  |
| --- | --- | --- |
|  | **CNN** | **ANN** |
| **Data** | Image data | Tabular data |
| **Recurrent connections** | No | No |
| **Parameter sharing** | Yes | No |
| **Spatial relationship** | Yes | No |
| **Vanishing and Exploding Gradient** | Yes | Yes |

**4.1.2 Pre-Processing**

In the step of pre-processing the main idea is to remove low-frequency background noise, normalizing the intensity of the images, and removing the reflections. Before extracting the features of any sign image in bad weather conditions pre-processing is used to improvise the image production.

**4.1.3 Data Representation**

In data representation, data can be represented in the form of binary numbers through which machine can understand data more clearly and effectively. Handling images of different sizes is done using this method. Going to the next part we would have classified the following: -

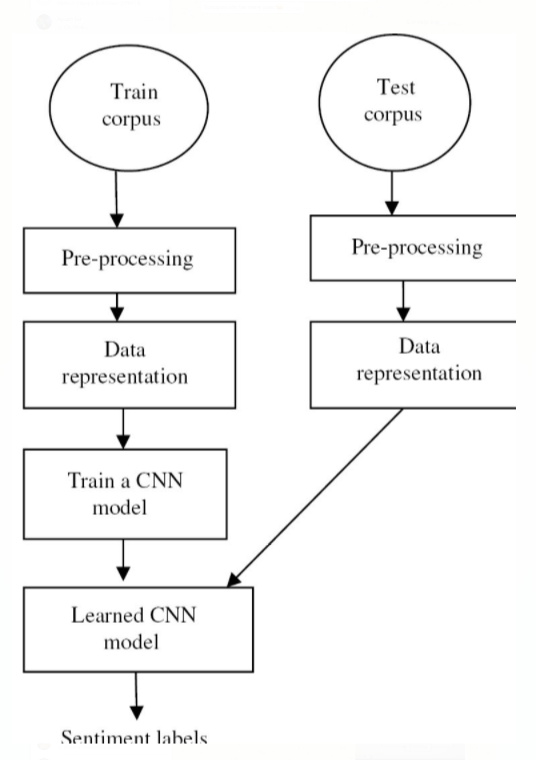
1. Blurring

2. Color based detection

3. Shape based detection

4. Cropped picture

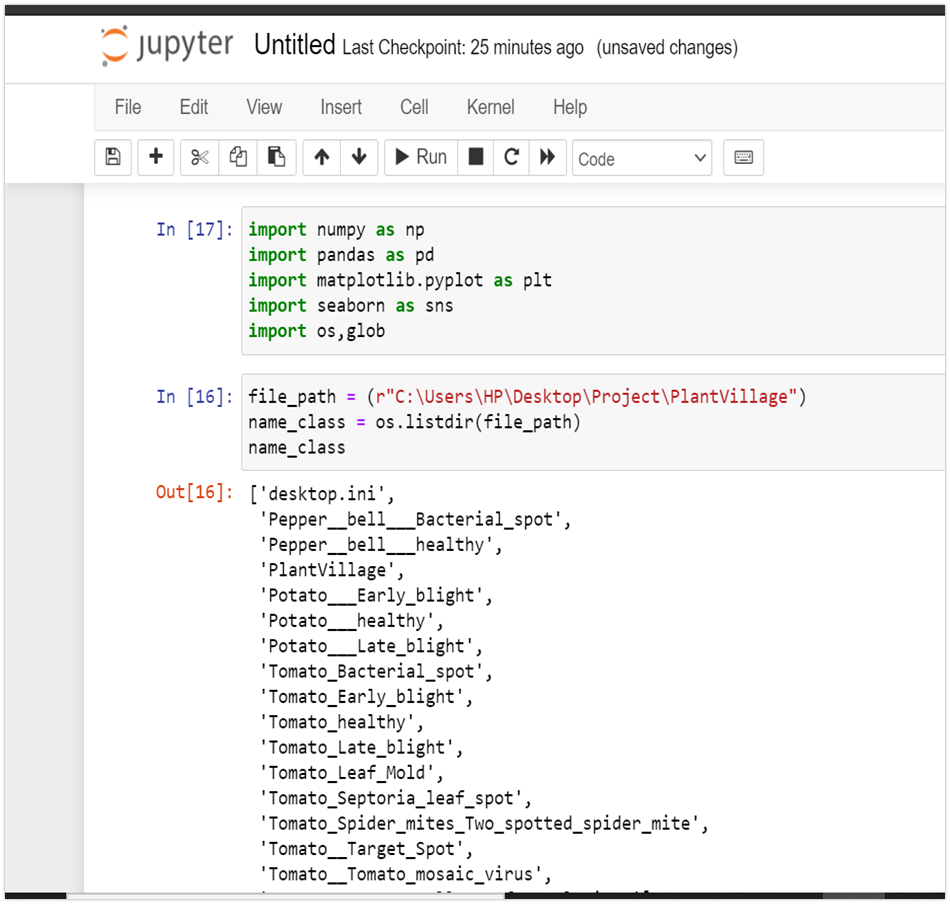
5. Separation



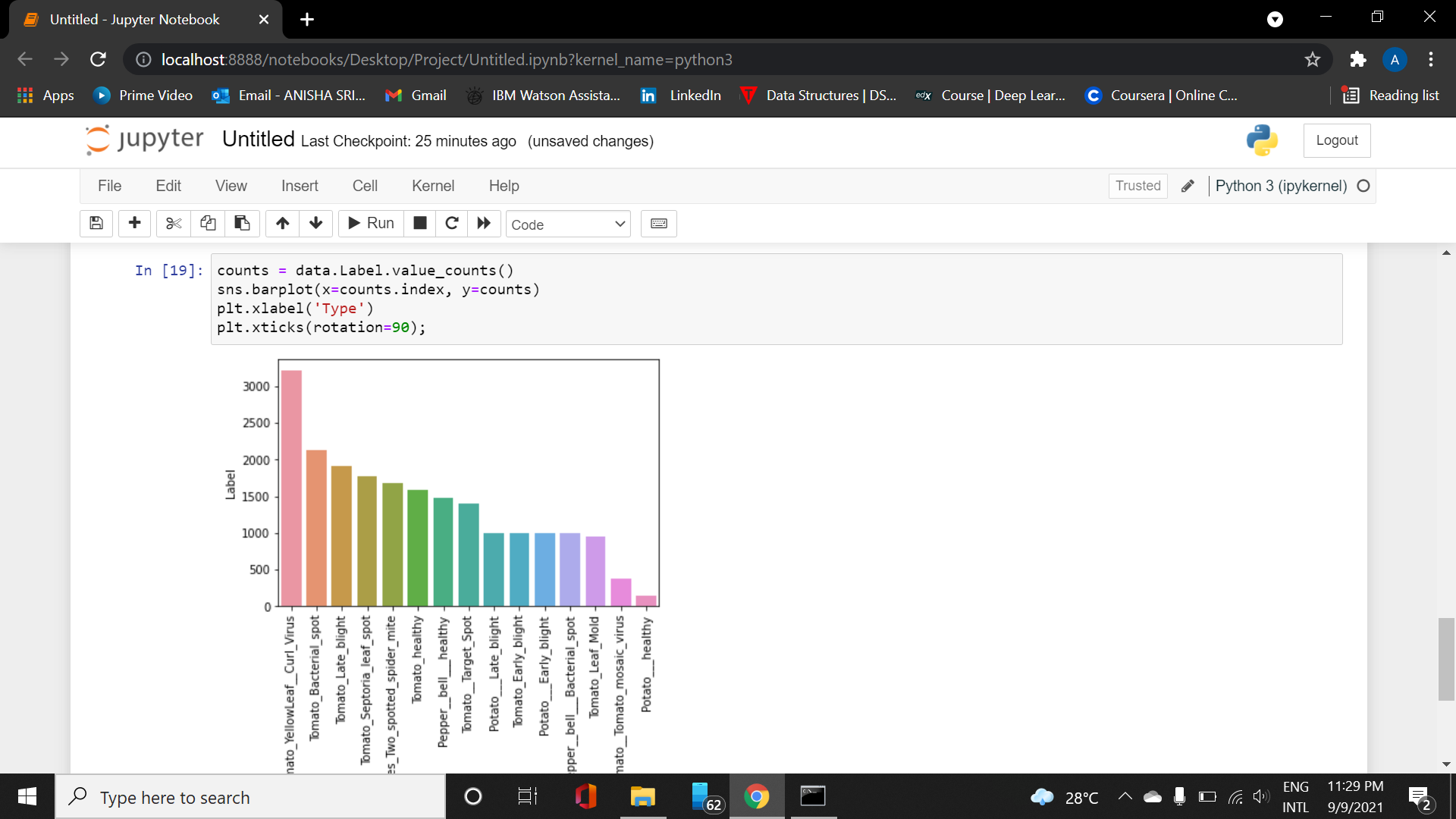
**Figure-4.1** Flow chart of the proposed model

**4.1.4 Training of CNN Model**

CNN (convolutional neural network) is an algorithm used in deep learning in which we can train our images inside the machine. It is most applied to analyzing visual imagery. CNN is a calculation which can take an information picture and can separate one from the another. CNN is used in plant disease prediction systems as it can identify a variety of diseases.



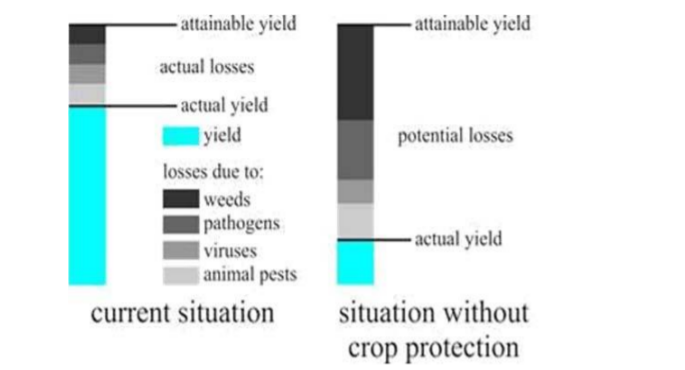
**Figure-4.2** Importing packages and creating dataset



**Figure-4.3** Training data present in dataset

**CHAPTER - 5 CONCLUSION AND ROAD MAP FOR PHASE-2**

**5.1 Conclusion**

Protecting plants is not an easy task. This depends on a thorough knowledge of the growing plant as well as insects, germs and weeds. In our program, a specialized in-depth learning model was developed based on a network of solutions to detect plant diseases with images of healthy or diseased plant leaves. Another feature that can be added to some programs is an intelligent program that treats diseases identified by plant-borne diseases. Various studies show that managing plant diseases can help increase yields by about 50%.****

**Fig 5.1.1 Crop loss because of disease**

The whole project is divided in three phases:

**Phase-1**

In the very first phase, we will be focussing on front-end of web application.

Also, two datasets will be created for training and testing of model.

**Phase-2**

In second phase, the whole CNN model will be created, and results will be generated.

**Phase-3**

In this phase, working of model will be ensured and web application will be able to predict plant disease by just uploading picture of diseased plant.

**Phase-1**

In the very first phase, we will be focussing on front-end of web application.

Also, two datasets will be created for training and testing of model.

**Phase-2**

In this phase the whole CNN model will be designed, and results will be generated

**Phase-3**

In this phase, working of model will be ensured and web application will be able to predict plant disease by just uploading picture of diseased plant.

and a comprehensive report will be developed including all three phases.

**Figure 5.1** - Phase wise project methodology

After completion of all the pre-processing steps the data is now ready for training and testing. Following is the plan of execution of phase 2

**4.1 Phase-2**

Figure 4.1 shows the flow diagram of phase 2. The CNN model will be implemented in phase 2. After model creation training and testing of model will be done on the datasets we have created in Phase-1. Finally, the accuracy of the model will be calculated by changing different parameters such as drop rate, learning rate, by changing epochs etc.



**Figure 4.1** Road map of phase 2

**BIBLIOGRAPHY**

1. Ying sun, Pingshu ge, Dequan Liu, “Traffic Sign Detection and Recognition Based on Convolutional Neural Network”, In proceedings of Chinese automation congress, Nov 2019.
2. Chen Li, Cheng Yang, “The Research on traffic sign recognition based on deep learning”, In proceedings of 16th International Symposium on Communications and Information Technologies, 2016.
3. Hao Xu, Gautam Srivastava, “Automatic recognition algorithm of traffic signs based on convolutional neural networks”, In proceedings of Multimedia Tools and Applications, Jan 2020.
4. Kwangyong Lim, Yongwon Hong, Yeongwoo Choi, Hyeran Byun, “Real-time traffic recognition based on general purpose GPU and deep-learning”, In proceedings of Xiaolie Ma, Beihang University, China, March 2017.
5. Mykola, “GTSRB-German Traffic Sign Recognition Benchmark”, In proceedings of International Joint Conference on Neural Networks (IJCNN), Nov 2018.
6. An Intuitive Explanation of Convolutional Neural Networks, Accessed on Oct 2020, <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
7. Deep learning vs machine learning: a simple way to understand the difference, Accessed on Oct 2020.

<https://www.zendesk.com/blog/machine-learning-and-deep-learning/#:~:text=To%20recap%20the%20differences%20between,intelligent%20decisions%20on%20its%20own>

1. Wes McKinney, Python for Data Analysis, Shroff Publishers & Distributors Pvt. Ltd, Third Edition, Indian Reprint, July 2015.
2. Wesley J Chun, Core Python Application Programming, Third Edition, Pearson.
3. Reema Thareja, Python Programming Using Problem Solving Approach, Fourth Edition, Oxford University Press.
4. Sebastian Raschka & Vahid Mirjalili, Python Machine Learning, Second Edition, Packt.